ANL-75-79

[redacted]

ANL-75-79

IMPROVED ASP I/O PERFORMANCE

by

Douglas E. Engert

ARGONNE NATIONAL LABORATORY, ARGONNE, ILLINOIS

Prepared for the U. S. ENERGY RESEARCH
AND DEVELOPMENT ADMINISTRATION
under Contract W-31-109-Eng-38

Printed in the United States of America
Available from
National Technical Information Service
U. S. Department of Commerce
5285 Port Royal Road
Springfield, Virginia 22161
Price: Printed Copy $4.50; Microfiche $2.25

Distribution Category:
Mathematics and
Computers (UC-32)

ANL-75-79

ARGONNE NATIONAL LABORATORY
9700 South Cass Avenue
Argonne, Illinois 60439

IMPROVED ASP I/O PERFORMANCE

by

Douglas E. Engert

Applied Mathematics Division

November 1975

TABLE OF CONTENTS

LIST OF FIGURES

IMPROVED ASP I/O PERFORMANCE

by

Douglas E. Engert

ABSTRACT

An improvement to the I/O routines of ASP is described, which can save 25% of the EXCPs issued by ASP. Unused buffers in the buffer pool are used to retain parts of the single track table, and only two modules are changed.

The Problem

It has long been known that the biggest bottleneck in most ASP systems is the ASP Queue. Here at Argonne, we have recently made an improvement to the ASPIO routines in the module IONUC to lessen the load on the queue packs.

Currently, we are running ASP on a 4 Megabyte, 370/195 with an ASP region of 738K, four queue packs and 10 initiators. Only one cylinder is used on the first queue pack which is allocated to the Single Track Table (STT), but it does extend to the other queue packs as well. Using software monitors, we have measured the overall activity on the queue packs and have found it to be as high as 70 interrupts per second. The activity is spread very evenly over all four of the queue packs, even though the first pack uses only one cylinder. Needless to say, the queue packs and the system residence volume are the busiest packs in the system. Any improvement in the area of ASPIO would be of major significance.

After noting the heavy load placed on the system by the Single Track Table records, a study was made to determine how the STT is used and what changes could be made.

Background

The STT is used for non-job related records, and for certain DSP's with control blocks of short duration, or only a few records. In these cases, the use of a half cylinder (the normal allocation unit for queue space) would be wasteful. Job related records such as the Job Control Table (JCT) and Network Control Blocks (NCB) also use the STT. The most active of these is the JCT. With a 250 job queue, typical for our system, this table occupies about 40 records, which are constantly being updated.

Four macros are used to perform I/O operations on records in the Single Track Table. AREAD is used to read a record. The track address of the record to be read is supplied to ASPIO. AWRITE is used to write a record. The address of the buffer to be written is supplied. The first four bytes of the record contain the track address. If they are zero, a new track address is obtained. ARELEASE is used to free the buffer used to hold a record, if it does not need to be rewritten. The buffer address is supplied. APURGE is used to free a record and its track address so it can be used for other functions.

Buffers for I/O are normally obtained from the ASP buffer pool. When the pool is empty, buffers are obtained from free storage. We have specified 115 buffers in our pool, of which usually 70 to 80 are in use. Since there has been no real benefit in increasing the size of the pool, we have tried to keep the size down. This has changed in light of this performance change.

An example of the use of a Single Track Table is when a card reader function is called. This occurs for each batch of jobs read in through RJP or our local RADS system. A command such as *X,CR,IN=R013 is issued to CALLDRVR. This function makes up a Job Description Accounting Block (JDAB) and a parameter buffer containing "IN=R013" and issues AWRITE macros for both

of these to write them to the queue. A JCT entry is then created. The routines that handle the JCT (JOBCONTL) issue AREAD macros to read in part of the JCT, to add the new entry. The updated JCT is then written to the queue using the AWRITE macro. CALLDRVR then posts the job segment scheduler (JSS). JSS determines that there is work to do, so it issues AREAD macros to read in parts of the JCT to look for work. It soon finds the function that needs to be scheduled, creates the incore tables needed, and posts it. JSS then updates the JCT entry to show that the card reader is active and issues an AWRITE macro to save this information. The card reader, which is now an active function, issues AREAD macros to read its JDAB and parameter buffer to find "IN=R013". ARELEASE macros are then issued for both of these records. When the card reader is finished, these same types of operations are performed again by JSS to update the JCT to mark the card reader function complete. These same types of actions are performed many times for all jobs in the system, making the JCT the heaviest used table residing on the queue packs.

In the above example, the JDAB and parameter buffer were read once, but the JCT was accessed twice. Depending on the size of the queue, the card reader entry may not be in the first record of the JCT, so other records may also have to be read. This JCT entry may be in the second or third record, so as many as six records may have to be read. ARELEASE macros are issued for JCT records that are read but not changed.

The Solution

This problem of high activity on the JCT has been approached by other installations, also. One approach was to keep an incore table of some of the JCT information so inquiries and quick searches for jobs could be made.

Only 70% of our buffer pool is in use. If the other 30% could be utilized to save some of the Single Track Table records, then they would not have to be read in each time.

Our approach at Argonne was to keep the changes localized to the ASPIO routine, IONUC, by modifying the AREAD, ARELEASE and AWRITE routines. This was deliberately done so the code could be exported easily. In this way, all Single Track Table activity could be improved. In the above example, it would be possible to save all the AREADS for the JCT, the JDAB, and the parameter buffer, since the card reader is normally scheduled in a fraction of a second.

When an AWRITE for a STT record is performed, instead of just returning the buffer to the buffer pool, it is added to a chain of incore retained buffers. The word preceding each buffer is used as a chain field. This word is normally used for debugging. The record can be identified since its track address is contained in the first four bytes.

When an AREAD for a STT record is performed, this chain of retained buffers is searched. If the record is found, it is dechained. The normal ASPIO activity is bypassed. This saves one I/O operation.

ARELEASE also will add STT records to the retained buffer chain, but one piece of additional information must be supplied. When an AWRITE is issued, there is no doubt that the incore data is the same as the data on the queue pack, since it was just written. When an ARELEASE is issued, there is no guarantee that the record has not been altered or that this is

the same record as on the queue pack. To supply this information, a new parameter has been added to the ARELEASE macro, TYPE=REAL. The logic of the function issuing the ARELEASE must determine if this is a copy of the record on the queue pack and pass this information along to the ARELEASE function. Currently, this is only done by JOBCONTL.

Whenever a new track is allocated, the list of retained buffers is checked; if a match is found, the buffer is removed from the chain. This must be done since an APURGE could have been issued for this track address. This is done in IONUC rather than when the APURGE is issued to localize the changes.

Code was also added to the GETBUF routine. When no more buffers are available from the buffer pool, GETBUF will take the first retained buffer. Since the buffers are added at the end, this will keep the most recently used records in core. Under the worst case, when all the buffers are in use and no buffers are retained, this algorithm will function the same as the unmodified system.

After a month of experience with this change, another parameter was added. A minimum number of retained buffers was set to 10. GETBUF would not use retained buffers for non-single track requests if the number of retained buffers was less than the minimum. This change was needed, since MAINIO could burst large amounts of output and use all the buffers. Still later, this limit was changed from 10 to 30.

Counters were added later to record the STT activity. The number of STT AREAD macros issued, the number of I/O operations saved by finding the record in a retained buffer, the number of buffers retained, the number of buffers used by GETBUF for other uses, and the number of EXCPs issued were recorded.

The Evaluation

Data was collected on a number of days during the day and early evening when we have our heaviest loads. The counters were recorded every 15 to 30 minutes using the Dump Core function (DC). The number of events per second for the five counters were calculated over the 15 to 30 minute time interval and were plotted. The hit ratio, or percent of STT AREADs requiring no I/O, and the percent of EXCPs saved were also plotted.

June 5, the first day this change was on the system, we hit an all time record high for the number of jobs run! As expected, during the afternoon when the load is the heaviest, the number of buffers used by GETBUF increased; so fewer buffers were retained, and the hit ratio dropped to about 20%. This is still a saving of about 6% in overall queue pack activity. Earlier the same day, the hit ratio hit 95% for an overall saving of queue pack activity of 25%.

On July 10, the minimum number of retained buffers parameter was added and set to 10. This was done to keep MAINIO from using all the buffers. The system performed about the same as on June 5.

Then on August 27, twenty buffers were added to the buffer pool; and the minimum number of retained buffers was set to 30. Figures 1 through 4 show the results of that change. During the day, the system went down three times at about 10:45, 15:45 and 16:45. This is rather unusual; but the data was used anyway, since it was a busy day and the counters are difficult to record using DC. Figure 3 shows the hit ratio, or percent of STT AREADs saved, to be about 87% with an overall savings of about 25% in queue pack activity. This was with a 250 job queue.

Appendix B contains the listing of the program used to produce these graphs, along with the output listing for August 27.

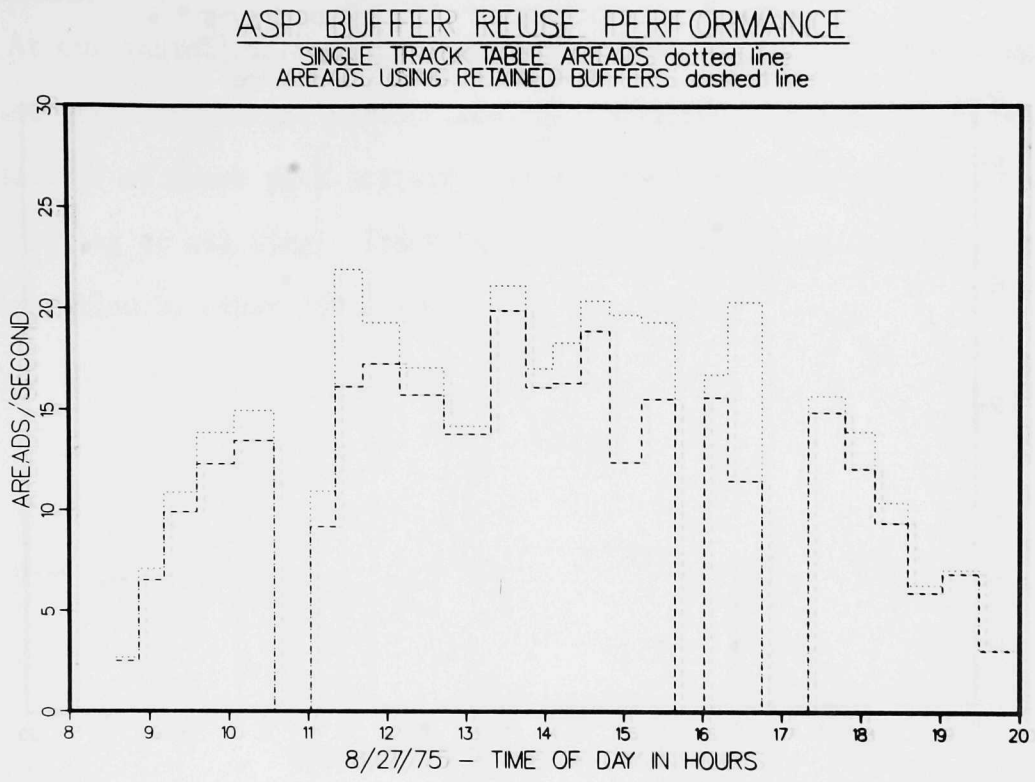

Figure 1

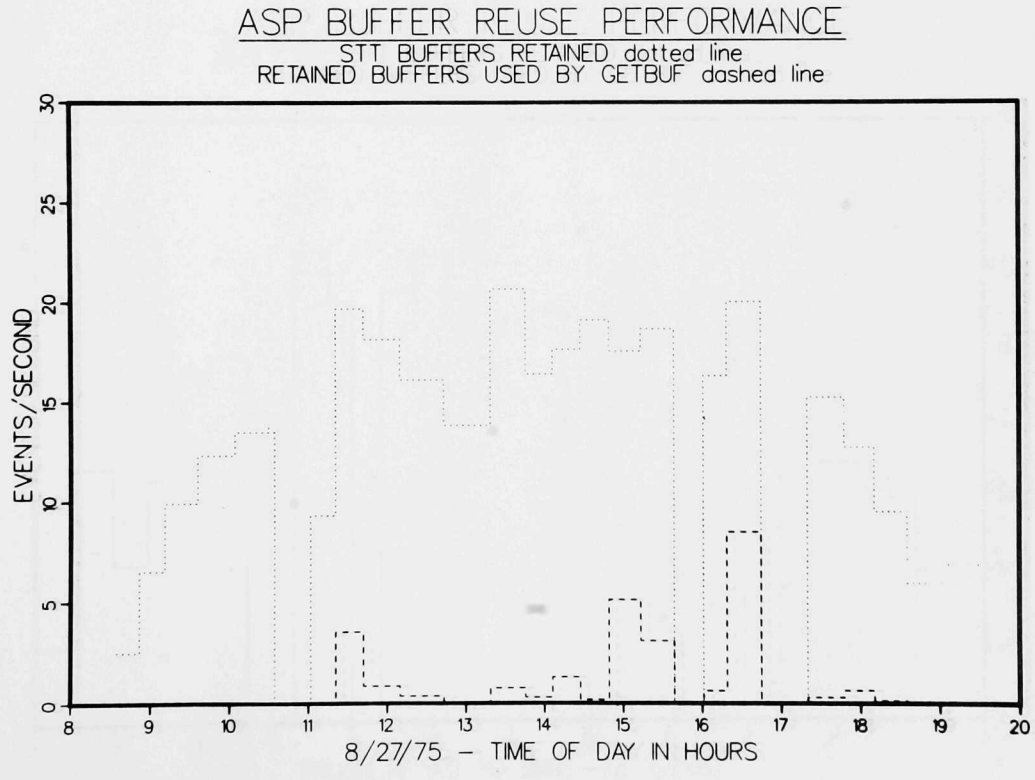

Figure 2

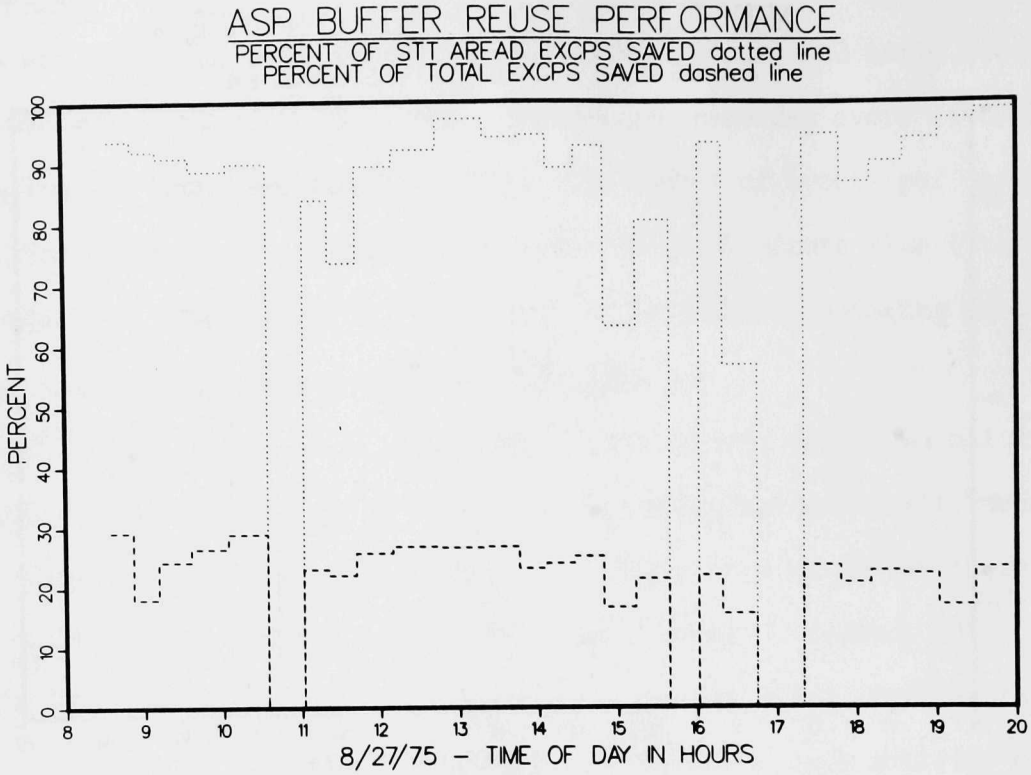

Figure 3

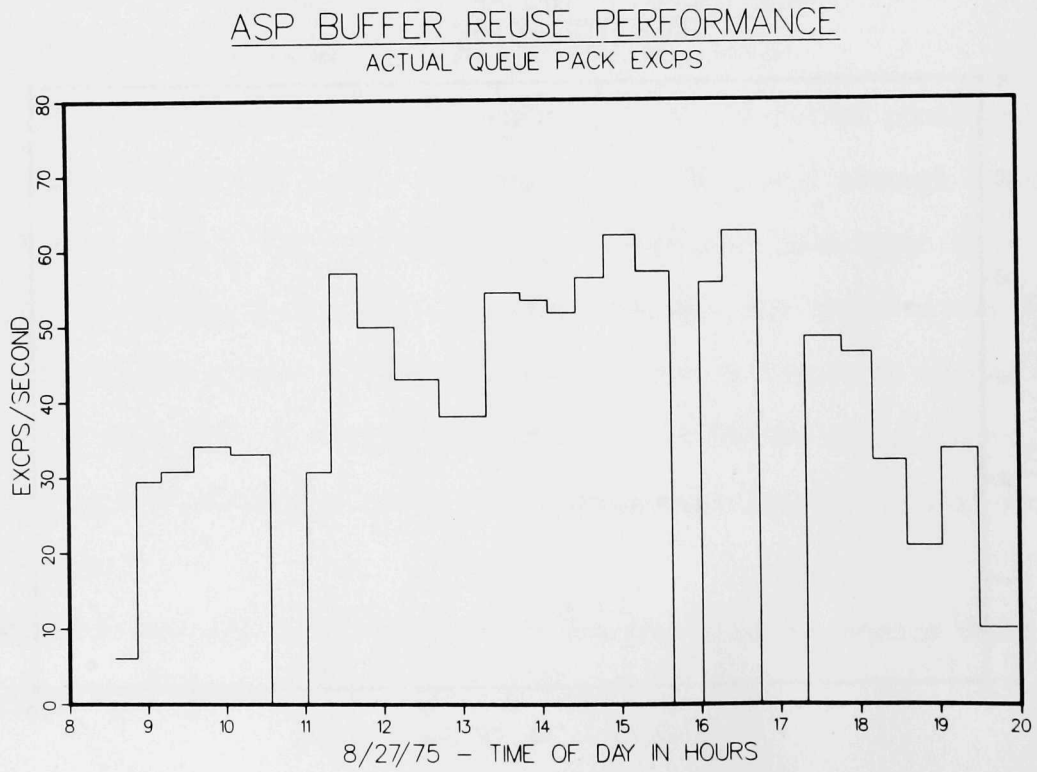

Figure 4

Conclusion

At our installation, this change has made a substantial improvement in channel activity and ASP performance. The benefits available: 1) upwards of 25% savings of queue pack activity, 2) the localized code charges, and 3) the encompassing of all Single Track Table activity make this change worthy of consideration by other ASP users.

Appendix A

The following changes are for ASP 3.1.3.

```
./        CHNGE ARELEASE,00,0,0
&NAME     ARELEASE &FDB=,&SAVE=,&NORMAL=,&TYPE=                    A04103 00000100
          AIF   ('&TYPE' NE 'REAL').A1                             A04103 00000210
          L     R15,ARELEASE -      ENTRY POINT                    A04103 00000211
          O     R15,=XL4'80000000' - SET TYPE=REAL                 A04103 00000212
          ZCALL (R15),SAVE=&SAVE,NORMAL=&NORMAL                    A04103 00000213
          MEXIT                                                    A04103 00000214
.A1       ANOP                                                     A04103 00000215


./        CHNGE ZCALL,00,0,0
&NAME     ZLOAD (R15),TVADD,&TVADD,L                               A04103 00003600


./        CHNGE IONUC,00,0,0
          TM    FDBFLAGS,FDBMNTAT   MAIN TAT?                      A04103 00208010
          BC    ALLOFF,ANLRD050     NO,BRANCH                      A04103 00208020
          TM    FDBFLAG0,FDBJBTAT   TAT?                           A04103 00208030
          BC    ALLON,ANLRD050      YES,BRANCH                     A04103 00208040
          LA    R15,ANLRD000                                       A04103 00208100
          BR    R15                 GO SEE IF BUFFER IS IN CORE    A04103 00208200
ANLRD050  EQU   *                                                  A04103 00208400
GETBF20   L     R5,ANLBFCHN         POINT AT BUFFER CHAIN          A04103 00594000
          LTR   R5,R5               ANY?                           A04103 00594010
          BC    ZERO,ANLGB010       NO,BRANCH                      A04103 00594020
          LH    R0,ANLCHCUR         CURRENT NUM OF BUFS ON CHAIN   A04103 00594021
          CLC   =AL3(ANLRDSIN),GETBFR14+1 SINGLE REC INPUT?        A04103 00594022
          BC    NE,ANLGB002         NO,BRANCH                      A04103 00594023
          TM    FDBFLAGS,FDBMNTAT   MAIN TAT?                      A04103 00594024
          BC    ALLON,ANLGB004      YES,USE A RETAINED BUFFER      A04103 00594025
ANLGB002  CH    R0,ANLCHMIN         MIN NUMBER?                    A04103 00594026
          BC    LE,ANLGB010         YES,DO A GETMAIN INSTEAD       A04103 00594027
ANLGB004  BCTR  R0,0                -1                             A04103 00594028
          STH   R0,ANLCHCUR         SAVE                           A04103 00594029
          LA    R0,1                                               A04103 00594030
          A     R0,ANLCHGB          INC COUNT OF BUFFERS NOT USED  A04103 00594040
          ST    R0,ANLCHGB                                         A04103 00594050
          L     R0,0(R5)            NEXT BUFFER ON CHAIN           A04103 00594060
          LTR   R0,R0                                              A04103 00594070
          BC    NZERO,ANLGB005      YES,BRANCH                     A04103 00594080
          MVC   ANLBFCHE,=A(ANLBFCHN) CHANGE END PTR ALSO          A04103 00594090
ANLGB005  ST    R0,ANLBFCHN         POINT TO NEXT                  A04103 00594100
          MVI   0(R5),C'B'          SET POOL BUFFER AGAIN          A04103 00594110
          LA    R0,4(R5)            POINT AT BUFFER                A04103 00594120
          B     GETBF05                                            A04103 00594130
ANLGB010  LH    R0,AIOBUFS4         SIZE FOR GETMAIN               A04103 00594140
          CLI   0(R14),C'D'         CHAIN BUFFER?                  A04103 00656010
          BC    NE,ANLPB010         NO,BRANCH                      A04103 00656020
```

```
         L     R5,ANLBFCHE          POINT AT END                    A04103 00656030
         ST    R14,0(R5)            SAVE END POINTER                A04103 00656040
         ST    R14,ANLBFCHE                                         A04103 00656050
         MVC   0(4,R14),TVTZERO     ZERO CHAIN FIELD                A04103 00656060
         LA    R14,1                                                A04103 00656070
         A     R14,ANLCHPB          INC COUNT OF OUR PUTBUFS        A04103 00656080
         ST    R14,ANLCHPB                                          A04103 00656090
         LA    R14,1                                                A04103 00656092
         AH    R14,ANLCHCUR         CURRENT # OF BUFFS ON CHAIN     A04103 00656094
         STH   R14,ANLCHCUR                                         A04103 00656096
         B     PUTBF25                                              A04103 00656100
ANLPB010 EQU   *                                                   A04103 00656110
         LA    R15,0(R15)           MAKE NOT MINUS                  A04103 00761100
         LTR   R15,R15              REAL BUFFER?                    A04103 00770010
         BC    NMINUS,ANLRE010      NO,BRANCH                       A04103 00770020
         SH    R2,=H'4'             -4                              A04103 00770030
         CLI   0(R2),C'C'           POOL BUFFER?                    A04103 00770040
         BC    NE,ANLRE010          NO,BRANCH                       A04103 00770050
         MVI   0(R2),C'D'           SET TO SAVE                     A04103 00770060
ANLRE010 EQU   *                                                   A04103 00770070
         L     R1,AIOBFDB           FDB                             A04103 00800100
./       DELET 00802000,00802000                                    A04103
         LA    R4,0(R4)                                             A04103 00819100
         SH    R4,=H'4'                                             A04103 00819200
         CLI   0(R4),C'D'           BUFFER TO BE RETAINED?          A04103 00819300
         BC    NE,ANLDK22           NO,BRANCH                       A04103 00819400
         OI    FDBFLAGS,FDBCLOSE    SET CLOSED                      A04103 00819500
ANLDK22  EQU   *                                                   A04103 00819600
ANLRDSIN EQU   *                                                   A04103 00949002
         BAL   R14,ANLCHKBF         CHECK IF THIS ADDRESS IS IN BUA04103 00970100
         BAL   R14,ANLCHKBF         CHECK IF THIS ADDRESS IS IN BUA04103 01031100
         BAL   R14,ANLCHKBF         CHECK IF THIS ADDRESS IS IN BUA04103 01046100
         BAL   R14,ANLCHKBF         CHECK IF THIS ADDRESS IS IN BUA04103 01141100
         LA    R14,1                ONE                             A04103 01191010
         A     R14,ANLCHIO          ADD EXCP COUNT                  A04103 01191020
         ST    R14,ANLCHIO          SAVE                            A04103 01191030
ANLCHKBF L     R15,ANLBFCHN         LOAD CHAIN OF BUFFERS           A04103 01269010
ANLCH010 LTR   R15,R15              ANY?                            A04103 01269020
         BCR   ZERO,R14             NO,RETURN                       A04103 01269030
         C     R0,4(R15)            THIS TA?                        A04103 01269040
         BC    EQ,ANLCH020          YES,BRANCH                      A04103 01269050
         L     R15,0(R15)           NEXT                            A04103 01269060
         B     ANLCH010             LOOK AT NEXT                    A04103 01269070
ANLCH020 XC    4(4,R15),4(R15)      CLEAR ADDRESS                   A04103 01269080
         BR    R14                                                  A04103 01269090
ANLBFCHN DC    A(0)                 CHAIN OF BUFFERS THAT MAY BE UA04103 01279010
ANLBFCHE DC    A(ANLBFCHN)          LAST BUFFER ON CHAIN            A04103 01279020
ANLCHRDS DC    F'0' COUNT OF AREADS THAT MAY BE SATISFIED          A04103 01279030
ANLCHRDN DC    F'0' COUNT OF AREADS THAT ARE BE SATISFIED          A04103 01279040
ANLCHPB  DC    F'0' COUNT OF PUTBUFS FOR OUR BUFFERS               A04103 01279050
ANLCHGB  DC    F'0' COUNT OF BUFFERS USED BY GETBUF                A04103 01279060
ANLCHIO  DC    F'0'                 EXCP COUNT                      A04103 01279065
ANLCHCUR DC    H'0'                 # OF BUFFS ON THE CHAIN         A04103 01279067
ANLCHMIN DC    H'30'                # OF RESERVED FOR RETAINED BUFA04103 01279069
```

```
         LTORG                                                          A04103 01279070
         USING ANLRD000,R15                                             A04103 01279100
         USING OPEN,R10                                                 A04103 01279110
ANLRD000 LA    R14,1                                                   A04103 01279120
         A     R14,ANLCHRDS       INC COUNT OF AREADS                  A04103 01279130
         ST    R14,ANLCHRDS                                            A04103 01279140
         LA    R14,ANLBFCHN       POINT AT BUFFER CHAIN                A04103 01279150
         L     R3,ANLBFCHN        FIRST BUFFER                         A04103 01279160
ANLRD010 LTR   R3,R3              ANY?                                 A04103 01279170
         BC    ZERO,ANLRD050      NO,BRANCH                            A04103 01279180
         CLC   FDBDATA,4(R3)      SAME ADDRESS?                        A04103 01279190
         BC    EQ,ANLRD020        YES,USE IT                           A04103 01279200
         LR    R14,R3                                                  A04103 01279210
         L     R3,0(R3)                                                A04103 01279220
         B     ANLRD010                                                A04103 01279230
ANLRD020 MVC   0(4,R14),0(R3)     SAVE CHAIN                           A04103 01279240
         CLC   0(4,R14),TVTZERO   LAST?                                A04103 01279250
         BC    NE,ANLRD030        NO,BRANCH                            A04103 01279260
         ST    R14,ANLBFCHE       SAVE END                             A04103 01279270
ANLRD030 ST    R1,0(R3)           CHAIN IN FDB                         A04103 01279280
         MVI   0(R3),C'C'         SET POOL BUFFER                      A04103 01279290
         LA    R3,4(R3)           POINT AT BUFFER                      A04103 01279300
         ST    R3,FDBDATA                                              A04103 01279310
         L     R9,AIOPARMS                                             A04103 01279320
         LH    R14,AIOBFUSE                                            A04103 01279330
         LA    R14,1(R14)                                              A04103 01279340
         STH   R14,AIOBFUSE                                            A04103 01279350
         CH    R14,AIONOBFM                                            A04103 01279360
         BC    LE,ANLRD040                                             A04103 01279370
         STH   R14,AIONOBFM                                            A04103 01279380
ANLRD040 LA    R14,1                                                   A04103 01279390
         A     R14,ANLCHRDN       INC COUNT OF READS SATISFIED         A04103 01279400
         ST    R14,ANLCHRDN                                            A04103 01279410
         LH    R14,ANLCHCUR                                            A04103 01279420
         BCTR  R14,0              -1                                   A04103 01279430
         STH   R14,ANLCHCUR       SAVE CURRENT # OF BUFFS ON CHA       A04103 01279440
         LR    R14,R6                                                  A04103 01279450
         B     CLSAWAIT                                                A04103 01279460
         DROP  R10                                                     A04103 01279470
         DROP  R15                                                     A04103 01279480
         TM    FDBFLAG0,FDBNOPUT+FDBJBTAT NO PUT BUF OR TAT?           A04103 01307010
         BC    NALLOFF,ANLCHE20   YES,BRANCH                           A04103 01307020
         TM    FDBFLAGS,FDBMNTAT  MAIN TAT?                            A04103 01307030
         BC    ALLOFF,ANLCHE20    NO,BRANCH                            A04103 01307040
         L     R12,AIOBSIOC                                            A04103 01307050
         CLI   0(R12),X'23'       SET SECTOR?                          A04103 01307060
         BC    NE,ANLCHE10        NO,BRANCH                            A04103 01307070
         LA    R12,8(R12)                                              A04103 01307080
ANLCHE10 L     R12,16(R12)        LOAD BUFFER ADDR                     A04103 01307090
         SL    R12,=F'4'          BACKUP 4                             A04103 01307100
         CLI   0(R12),C'C'        POOL BUFFER?                         A04103 01307110
         BC    NE,ANLCHE20        NO,BRANCH                            A04103 01307120
         MVI   0(R12),C'D'        SET CHAIN                            A04103 01307130
         B     CHEND12                                                 A04103 01307140
ANLCHE20 OI    FDBFLAGS,FDBCLOSE  SET CLOSED                           A04103 01307150
         BC    ALLOFF,CHEND80                                          A04103 01310000
./       CHNGE JOBCONTL,00,0,0
TAGET180 ARELEASE FDB=TAJCURFD,TYPE=REAL                               A04103 00218000
         ARELEASE FDB=TAJCURFD,NORMAL=TAEXIT0,TYPE=REAL                A04103 00339000
DEQ05    ARELEASE FDB=TAJCURFD,NORMAL=(R8),TYPE=REAL                   A04103 00803000
```

Appendix B

The following Fortran program was used to analyze the data. The dump core dsp, DC, was used to obtain the counters. The plots are produced by the DISSPLA package from Integrated Software Systems Corporation.

```
C
C     INPUT FORMAT :
C
C     FIRST CARD HAS DATE AND BLANK FOR NO PLOT OR PLOT
C
C     ADD A CARD FOR EVERY RESTART WITH THE RESTART TIME AND THE
C     BACKLOG
C
C     DATA CARDS:
C
C      COL.    DEF.                        NAME      IONUC NAME
C
C       1- 6  TIME OF READING              TH,TM,TS
C       7-14  AREADS                       ARD       ANLCHRDS
C      15-22  AREADS FOUND                 ARDS      ANLCHRDN
C      23-30  BUFFERS RETAINED             PUTB      ANLCHPB
C      31-38  BUFFS USED BY GETBUF         GETB      ANLCHGB
C      39-43  BUFF POOL EXCEEDED (*I,C)    PE
C      44-51  EXCPS                        EXCP      ANLCHIO
C      52-56  BACKLOG (*I,B) R/I           BR
C      57-61  BACKLOG (*I,B) SETUP         BS
C      62-66  BACKLOG (*I,B) MAIN          BM
C      67-71  BACKLOG (*I,B) PERF          BP
C
C     THE PLOT ROUTINES ARE FOR THE DISSPLA PACKAGE
C     THEY MAYBE LEFT OUT IF THE PACKAGE IS NOT AVAILABLE
C
      INTEGER SWITCH,XTITLE(8)/'XX/XX/XX - TIME OF DAY IN HOURS$'/,
     X   PLUT/'PLOT'/
      INTEGER  TH(100),TM(100),TS(100),SEC(100),ARD(100)
      INTEGER  ARDS(100),GETB(100),PUTB(100),PE(100),ICURD(100)
      INTEGER EXCP(100),BR(100),BS(100),BM(100),BP(100)
      DIMENSION XH(100),YARD(100),YARDS(100),YPUTB(100),YGETB(100)
      DIMENSION YCUR(100),IPK(1000),YRAT(100),YRATT(100),YEXCP(100)
      DO 21 I=1,100
21    ICURD(I)=0
      READ(5,20)XTITLE(1),XTITLE(2),SWITCH
20    FORMAT(2A4,1X,A4)
      IF(SWITCH.NE.PLUT)GO TO 81
      CALL STRTPL
      CALL BGNPL(-1)
      CALL PAGE(11.0,12.45)
      CALL NOBRDR
      CALL SIMPLX
```

```
      CALL MIXALF('L/CSTD')
      CALL XINTAX
      CALL YINTAX
81    CONTINUE
      J=0
      I=0
1     I=I+1
      READ(5,2,END=3) TH(I),TM(I),TS(I),ARD(I),ARDS(I),PUTB(I),GETB(I),
     X PE(I),EXCP(I),BR(I),BS(I),BM(I),BP(I)
2     FORMAT(3I2,4Z8,I5,Z8,4I5)
      SEC(I)=(TH(I)*60+TM(I))*60+TS(I)
      WRITE(6,14)TH(I),TM(I),TS(I),I,ARD(I),ARDS(I),PUTB(I),GETB(I),
     X PE(I),EXCP(I),BR(I),BS(I),BM(I),BP(I)
14    FORMAT(2X,I2,1H:,I2,1H:,I2,7I8,4I6)
      ICURD(I)=PUTB(I)-ARDS(I)-GETB(I)
      IF(ARD(I).EQ.0)J=I-1
      IF(J.EQ.0)GO TO 1
      ARD(I)=ARD(I)+ARD(J)
      ARDS(I)=ARDS(I)+ARDS(J)
      PUTB(I)=PUTB(I)+PUTB(J)
      GETB(I)=GETB(I)+GETB(J)
      PE(I)=PE(I)+PE(J)
      EXCP(I)=EXCP(I)+EXCP(J)
      GO TO 1
3     I=I-1
      WRITE(6,13)XTITLE(1),XTITLE(2)
13    FORMAT(1H1,2X,2A4
     X/' INTERVAL          STT    STT    STT  GETBUF  GET  ',
     X  '  QUEUE  JOBS   BUFFS    %    % STT  QUEUE   STT ',
     X  ' AREADS   STT  GETBUF  GET  '/
     X '   END       SEC AREADS AREADS PUTBUF OF STT  MAINS',
     X  '  EXCPS   IN      ON   EXCPS  EXCPS  EXCPS AREADS',
     X  '  SAVED PUTBUF  /SEC   MAINS'/
     X '   TIME                  SAVED        BUFFS        ',
     X  '        SYSTEM  CHAIN  SAVED  SAVED   /SEC  /SEC ',
     X  '  /SEC   /SEC          /SEC '/)
12    FORMAT(//)
      L=I-1
      DO 6 J=1,L
      K=J+1
      INDEX=1
      GO TO 10
4     CONTINUE
      XH(J)=(SEC(K)+SEC(J))/7200.0
      YARD(J)=PARD
      YARDS(J)=PARDS
      YPUTB(J)=PPUTB
      YGETB(J)=PGETB
      YRAT(J)=RAT
      YRATT(J)=RATT
      YEXCP(J)=PEXCP
6     CONTINUE
      J=I-1
      IF(SWITCH.NE.PLUT)GO TO 82
C
C     PLOT 1
C
      CALL TITLE(' ',-1,XTITLE,100,
```

```
     1  'AREADS/SECOND$',100,8.,5.)
      CALL HEADIN ('ASP BUFFER REUSE PERFORMANCE$',-100,3,3)
      CALL HEADIN('SINGLE TRACK TABLE AREADS (DOTTED LINE)$',100,2,3)
      CALL HEADIN('AREADS USING RETAINED BUFFERS (DASHED LINE)$',
     X  100,2,3)
      CALL GRAF(8.,1.,20.,0.,5.,30.)
      CALL FRAME
      CALL STEP
      CALL DOT
      CALL CURVE(XH,YARD,J,0)
      CALL DASH
      CALL CURVE(XH,YARDS,J,0)
      CALL RESET('DASH')
      CALL MESSAG('F(IGURE) 1$',100,3.5,-1.0)
      CALL ENDPL(1)
C
C     PLOT 2
C
      CALL TITLE(' ',-1,XTITLE,100,
     1  'EVENTS/SECOND$',100,8.,5.)
      CALL HEADIN ('ASP BUFFER REUSE PERFORMANCE$',-100,3,3)
      CALL HEADIN('STT BUFFERS RETAINED (DOTTED LINE)$',100,2,3)
      CALL HEADIN('RETAINED BUFFERS USED BY GETBUF (DASHED LINE)$',
     X   100,2,3)
      CALL GRAF(8.,1.,20.,0.,5.,30.)
      CALL FRAME
      CALL STEP
      CALL DOT
      CALL CURVE(XH,YPUTB,J,0)
      CALL DASH
      CALL CURVE(XH,YGETB,J,0)
      CALL RESET('DASH')
      CALL MESSAG('F(IGURE) 2$',100,3.5,-1.0)
      CALL ENDPL(2)
C
C     PLOT 3
C
      CALL TITLE(' ',-1,XTITLE,100,
     X  'PERCENT$',100,8.,5.)
      CALL HEADIN ('ASP BUFFER REUSE PERFORMANCE$',-100,3,3)
      CALL HEADIN('PERCENT OF STT AREAD EXCPS SAVED (DOTTED LINE$',
     X   100,2,3)
      CALL HEADIN('PERCENT OF TOTAL EXCPS SAVED (DASHED LINE)$',
     X   100,2,3)
      CALL GRAF(8.,1.,20.,0.,10.,100.)
      CALL FRAME
      CALL STEP
      CALL DOT
      CALL CURVE(XH,YRAT,J,0)
      CALL DASH
      CALL CURVE(XH,YRATT,J,0)
      CALL RESET('DASH')
      CALL MESSAG('F(IGURE) 3$',100,3.5,-1.0)
      CALL ENDPL(3)
C
C     PLOT 4
C
      CALL TITLE(' ',-1,XTITLE,100,
```

```
     X  'EXCPS/SECOND$',100,8.,5.)
      CALL HEADIN ('ASP BUFFER REUSE PERFORMANCE$',-100,3,2)
      CALL HEADIN('ACTUAL QUEUE PACK EXCPS$',100,2,2)
         CALL GRAF (8.,1.,20.,0.,10.,80.)
      CALL FRAME
      CALL STEP
      CALL CURVE(XH,YEXCP,J,0)
      CALL MESSAG('F(IGURE)  4$',100,3.5,-1.0)
      CALL ENDPL(4)
      CALL DONEPL
82    CONTINUE
      INDEX=2
      WRITE(6,12)
      J=1
      K=I
      GO TO 10
5     CONTINUE
      STOP
10    ISEC=SEC(K)-SEC(J)
      IARD=ARD(K)-ARD(J)
      IARDS=ARDS(K)-ARDS(J)
      IGETB=GETB(K)-GETB(J)
      IPUTB=PUTB(K)-PUTB(J)
      IEXCP=EXCP(K)-EXCP(J)
      IJOBS=BR(K)+BS(K)+BM(K)+BP(K)
      IPE=PE(K)-PE(J)
      ICUR=ICURD(K)
      RAT=0.0
      RATT=0.0
      IF(IARD.NE.0)RAT=IARDS*100.0/IARD
      IF(IEXCP+IARDS.NE.0)RATT=IARDS*100.0/(IEXCP+IARDS)
      PARD=IARD*1.0/ISEC
      PARDS=IARDS*1.0/ISEC
      PPUTB=IPUTB*1.0/ISEC
      PGETB=IGETB*1.0/ISEC
      PPE=IPE*1.0/ISEC
      PEXCP=IEXCP*1.0/ISEC
      WRITE (6,11) TH(K),TM(K),TS(K),ISEC,IARD,IARDS,IPUTB,IGETB,IPE,
     X  IEXCP,IJOBS,ICUR,RATT,RAT,PEXCP,PARD,PARDS,PPUTB,PGETB,PPE
11    FORMAT(1X,I2,1H:,I2,1H:,I2,5I7,I6,I8,2I7,2F7.1,6F7.2)
      GO TO (4,5            ),INDEX
      END
```

8/27/75

INTERVAL END TIME	SEC	STT AREADS	STT AREADS SAVED	STT PUTBUF	GETBUF OF STT BUFFS	GET MAINS	QUEUE EXCPS	JOBS IN SYSTEM	BUFFS ON CHAIN	% EXCPS SAVED	% STT EXCPS SAVED	QUEUE EXCPS /SEC	STT AREADS /SEC	AREADS SAVED /SEC	STT PUTBUF /SEC	GETBUF /SEC	GET MAINS /SEC
8:53:58	1298	3471	3256	3263	0	0	7882	86	49	29.2	93.8	6.07	2.67	2.51	2.51	0.0	0.0
9: 7:11	793	5616	5172	5184	30	132	23365	97	31	18.1	92.1	29.46	7.08	6.52	6.54	0.04	0.17
9:35:41	1710	18618	16941	16966	27	321	52655	109	29	24.3	91.0	30.79	10.89	9.91	9.92	0.02	0.19
10: 5:26	1785	24736	21961	21998	34	480	60914	132	32	26.5	88.8	34.13	13.86	12.30	12.32	0.02	0.27
10:30:47	1521	22753	20475	20507	32	325	50268	152	32	28.9	90.0	33.05	14.96	13.46	13.48	0.02	0.21
11:10:10	2363	0	0	0	0	0	0	206	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11:15:58	348	3808	3197	3237	0	0	10663	206	40	23.1	84.0	30.64	10.94	9.19	9.30	0.0	0.0
11:41: 8	1510	33133	24386	29707	5330	4270	86078	221	31	22.1	73.6	57.01	21.94	16.15	19.67	3.53	2.83
12: 9:17	1689	32616	29176	30633	1448	5294	84157	249	40	25.7	89.5	49.83	19.31	17.27	18.14	0.86	3.13
12:39:28	1811	30938	28503	29165	657	1438	77680	225	45	26.8	92.1	42.89	17.08	15.74	16.10	0.36	0.79
13:24:42	2714	38510	37437	37534	108	560	103041	228	34	26.6	97.2	37.97	14.19	13.79	13.83	0.04	0.21
13:46:10	1288	27219	25633	26611	972	1168	69980	232	40	26.8	94.2	54.33	21.13	19.90	20.66	0.75	0.91
14: 5: 1	1131	19271	18217	18545	331	1713	60330	231	37	23.2	94.5	53.34	17.04	16.11	16.40	0.29	1.51
14:28:29	1408	25803	22985	24815	1832	3112	72764	244	35	24.0	89.1	51.68	18.33	16.32	17.62	1.30	2.21
14:47:11	1122	22890	21218	21417	201	342	63171	247	33	25.1	92.7	56.30	20.40	18.91	19.09	0.18	0.30
15:13:58	1607	31668	19938	28181	8236	5985	99596	252	40	16.7	63.0	61.98	19.71	12.41	17.54	5.13	3.72
15:38:25	1467	28379	22800	27351	4551	4388	83840	249	40	21.4	80.3	57.15	19.34	15.54	18.64	3.10	2.99
16: 4:42	1577	0	0	0	0	0	0	289	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
16:16:34	712	11930	11122	11602	449	396	39669	289	31	21.9	93.2	55.71	16.76	15.62	16.29	0.63	0.56
16:37:45	1271	25854	14601	25415	10815	4840	79502	305	30	15.5	56.5	62.55	20.34	11.49	20.00	8.51	3.81
17:27:52	3007	0	0	0	0	0	0	305	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
17:46:54	1142	17929	17000	17392	348	414	55452	319	44	23.5	94.8	48.56	15.70	14.89	15.23	0.30	0.36
18:10:37	1423	19832	17208	18128	925	571	66180	284	39	20.6	86.8	46.51	13.94	12.09	12.74	0.65	0.40
18:34:40	1443	15031	13562	13746	184	311	46531	259	39	22.6	90.2	32.25	10.42	9.40	9.53	0.13	0.22
19: 2:28	1668	10538	9910	9936	33	9	34989	234	32	22.1	94.0	20.98	6.32	5.94	5.96	0.02	0.01
19:27:58	1530	10826	10528	10539	14	23	51665	211	29	16.9	97.2	33.77	7.08	6.88	6.89	0.01	0.02
19:58:58	1860	5772	5703	5712	4	295	18877	207	34	23.2	98.8	10.15	3.10	3.07	3.07	0.00	0.16